Project no. IST-033576

XtreemOS

Integrated Project
BUILDING AND PROMOTING A LINUX-BASED OPERATING SYSTEM TO SUPPORT VIRTUAL ORGANIZATIONS FOR NEXT GENERATION GRIDS

XtreemOS: a Vision for a Grid Operating System

XtreemOS Technical Report # 4

Toni Cortes, Carsten Franke, Yvon Jégou, Thilo Kielmann, Domenico Laforenza,
Brian Matthews, Christine Morin, Luis Pablo Prieto, and Alexander Reinefeld

Report Registration Date: May 14, 2008

Version 1 / Last edited by Brian Matthews / May 14, 2008

Project co-funded by the European Commission within the Sixth Framework Programme		
Dissemination Level		
PU	Public	√
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Revision history:

Version	Date	Authors	Institution	Section affected, comments
1	30/04/08	Toni Cortes, Carsten Franke, Yvon Jégou, Thilo Kielmann, Domenico Laforenza, Brian Matthews, Christine Morin, Luis Pablo Prieto, and Alexander Reinefeld		Initial document
2	01/05/08	Brian Matthews		Polishing Review

XtreemOS: a Vision for a Grid Operating System

Toni Cortes[*], Carsten Franke[†], Yvon Jégou[‡],
Thilo Kielmann[§], Domenico Laforenza[¶], Brian Matthews[‖],
Christine Morin[‡], Luis Pablo Prieto[**] and Alexander Reinefeld[††]

1 Introduction

For many businesses, the ability to dynamically adapt to changing environments has become a key success factor. Therefore, many companies need to significantly increase their agility and cost efficiency to survive in this dynamic environment. In industrial sectors (e.g. aeronautics, chemistry, ...), and in the R&D and academic research domains (e.g. genomics, high energy physics, ...), it is now standard practice to engage in joint development programmes between organisations, a practice which critically affects the underlying information and communication infrastructure. For example, many applications consist of several executables that need to be started, managed and stopped in a coordinated fashion. Further, many applications use databases in the range of several Gigabytes for storing business related data. Furthermore, this includes a large proportion of interactive applications, which thus have higher requirements for the direct customer response times. Hence, businesses are searching for new technologies that overcome current limitations and allow them to execute their businesses in an effective manner by providing a high degree of adaptability.

Many enterprises are operating in a distributed fashion. Thus, the whole company is divided into several administrative domains. Further, many joint research and development programmes exploit resources spanning multiple administrative domains. In order to run the overall business effectively, the different locations must cooperate and dynamically adapt as a whole during changes. One of the

[*]Barcelona Supercomputing Center (BSC), Spain
[†]SAP, Germany
[‡]INRIA Rennes-Bretagne Atlantique
[§]Vrije Universiteit, Amsterdam, The Netherlands
[¶]Istituto di Scienza e Tecnologie dell'Informazione (CNR, Pisa), Italy
[‖]Science and Technology Facilities Council, UK
[**]Telefonica I+D (TID), Spain
[††]Konrad-Zuse-Zentrum fuer Informationstechnik Berlin (ZIB), Germany

main goals during this operation is the minimization of administration tasks. Furthermore, it is essential for enterprise to be able to execute legacy software within these environments without the need to modify or recompile the various system components.

One possible way to address some of these requirements is to use Grid technologies. The term Grid computing was introduced at the end of 90s by Foster and Kesselman; it was envisioned as *"an important new field, distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and, in some cases, high-performance orientation"* [10]. Defining Grids precisely has always been difficult but nowadays there is a general agreement that Grids are distributed systems enabling the creation of *Virtual Organizations (VOs)* [11]. By working within VOs, users can share, select, and aggregate a wide variety of geographically distributed resources, owned by different organizations, to solve large-scale computation and data intensive problems in science, engineering, and commerce. Those resources may include any kind of computational resources like supercomputers, storage systems, data sources, sensors, and specialized devices.

More recently researchers belonging at the European Network of excellence "CoreGrid"[1] reached an agreement on the following definition: a Grid is *"a fully distributed, dynamically reconfigurable, scalable and autonomous infrastructure to provide location independent, pervasive, reliable, secure and efficient access to a coordinated set of services encapsulating and virtualizing resources (computing power, storage, instruments, data, etc.) in order to generate knowledge"*. This is a more modern service-oriented vision of the Grid that stems from the conviction that in the mid-to-long term the great majority of complex software applications will be dynamically built by composing services, which will be available in an open market of services and resources. In this sense, the Grid will be conceived as a "world-wide cyber-utility" populated by cooperating services interacting in a complex and gigantic software ecosystem.

The "Cloud computing" model introduced in 2007 [24] considers the outsourcing of hardware and software to Internet providers. In this model, there is no need to bother with hardware acquisition and management: users just rent the services (or virtual machines) they need, and they only pay for effective resource usage. Cloud computing is mainly based on virtualization technology and grid computing software.

The XtreemOS [5] project started in 2006 follows yet another approach to the management of large and very dynamic grid systems: users logged into an XtreemOS box will transparently exploit VO-managed resources through the stan-

[1] <http://www.coregrid.net/>

dard POSIX interface.

While much work has been done to build Grid middleware on top of existent operating systems, little has been done to extend the underlying operating systems for enabling and facilitating Grid computing, for example, by embedding some important basic services or functionalities directly into the operating system kernel. In this light, XtreemOS aims to be a first European step towards the creation of a true open source operating system for Grid platforms. The XtreemOS operating system is based on Linux traditional general-purpose OS, extended as needed to support VOs, and to provide appropriate interfaces to the Grid OS services. In contrast to middleware approaches, XtreemOS is an operating system able to execute any kind of application, including unmodified existing applications.

The realization of this new Grid vision introduces new challenges: transparency for users and application developers, scalability, manageability, security, trust. In the rest of this paper, we consider some of the challenges and opportunities which are addressed by XtreemOS

2 Fundamental Concepts and their Instantiation in XtreemOS

In this section, we first discuss the fundamental concepts that have driven the design of XtreemOS. Then, we present the overall architecture of XtreemOS and its key design principles.

2.1 Fundamental Concepts

The design of XtreemOS has been guided by two fundamental concepts: transparency and scalability.

2.1.1 Transparency

Transparency in XtreemOS can be seen from two different points of view: the user and the application. We consider what we mean by transparency in each case.

Transparency for the User

A traditional user will have the feeling that is still working on a Linux machine. For instance, traditional Linux commands will be used instead of new ones: jobs (including legacy and grid-unaware applications) will be submitted like regular processes are launched today (just by writing the program name); checking the status of jobs will be done by using the traditional `ps` command; and so on. This means that XtreemOS will bring the Grid to standard Linux users. It is important to

mention that submitting a job to the Grid requires some Grid-parameters such as the resources needed. These parameters will be predefined by application developers, and/or learned automatically by XtreemOS. On the other hand, we will also allow Grid-aware users to define these Grid parameters themselves to offer maximum power to expert users.

Another way in which XtreemOS will make the Grid transparent to users is that there will be no limit in the kind of applications that will be supported. Unlike most Grid systems that only allow batch jobs, XtreemOS will allow interactive (both terminal and Xwindows) applications to be submitted to the Grid in the same way they are launched in a Linux box.

Regarding user sessions, XtreemOS will offer a mechanism that once the user has logged into the VO, all commands will become Grid commands and thus the user will not have to worry about any of the Grid aspects anymore. In other words, logging into XtreemOS will launch a Grid-aware shell that transparently takes care of all Grid-related issues.

Finally, XtreemOS will allow VO to be built to either share or isolate resources from the rest of the world. This will be defined by the administrator of the VO, and thus it will be transparent to users. For instance, in an environment where isolation is vital, it will be handled by the VO management and the user will run jobs in an isolated environment without having to worry further.

Application Transparency

XtreemOS will also make Grid executions transparent for applications, and application developers, . First, the job semantics will be very similar to the process semantics. This means that XtreemOS will offer a hierarchy of jobs (much in the same way as the Linux process hierarchy) and the same system calls (same interface) used to manage processes will be able to handle jobs (i.e. wait for a job, send signals to a job, etc.). Furthermore, to simplify the task of programming, XtreemOS will treat process within a job in the same way Linux treats threads within a process (thus the programmer will not need to learn new relationships).

Another way in which XtreemOS will make the Grid transparent to applications is that files will be stored in XtreemFS, using a Grid file system, and applications will be able to access these files regardless of their physical location using the POSIX interface and semantics.

In addition, XtreemOS will offer transparent fault tolerance to applications. If a resource fails, the application will recover automatically and transparently from the user.

Finally, XtreemOS will also make clusters of computers transparent to applications because each cluster in the system will offer a *single-system image* to the

applications (i.e. single login point, distributed shared memory, etc.)

2.1.2 Scalability

Scalability is a key property of the XtreemOS operating system. XtreemOS has potentially to deal with a large number of resources (millions of nodes) owned by different providers and located in different sites (possibly thousands of location/administrative domains). It will be used by a large number of users (thousands of users) executing altogether a very large number of applications of various kinds (data intensive versus compute intensive), some of them being large scale applications spanning multiple Grid nodes and requiring a large amount of resources (individual jobs may span thousands of nodes). Multiple virtual organizations may rely on a Grid infrastructure, each with its own members, administrator, resources and policies. Thus XtreemOS has to be able to scale to these large sizes. We consider some of these factors in more detail.

Heterogeneity XtreemOS has to deal with a large amount of heterogeneous Grid nodes (resources) interconnected by various wired or wireless networks (e.g. WAN, LAN, SAN). These networks are heterogeneous from the performance point of view (latency, bandwidth, jitter) and performance is variable depending on the load. Resources are heterogeneous from the hardware point of view. There are different kinds of Grid nodes from the point of view of computer architecture: individual PC, clusters, high-performance computers and mobile devices. Grid nodes are based on different processor types and have a different amount of local resources (memory, disk, number of processors and cores). Grid nodes are also heterogeneous from the software point of view. They may run a different version of an operating system and they may be configured in different ways, for instance using the same (or same version) of libraries. Finally, Grid nodes belonging to different sites are independently managed and there is no reason to assume that the administration policies would be the same in these sites.

XtreemOS must be capable of running on a wide variety of different platforms, ranging from powerful servers down to simple PDAs or mobile phones with only little computational power. Hence, there will be several flavors of XtreemOS, but all with the same consistent set of services and the same interfaces.

Dynamicity Grid nodes may join or leave the system at anytime based on decisions of their administrator or user. This may happen with a prior announcement, via a sign on/off, or without, for example a crash. A given Grid node may be temporarily disconnected as networks failures may occur at any time. Some Grid

nodes, such as laptops, PDA, or mobile phones may suddenly be disconnected and later be re-joined with different data.

VO Models and Dynamicity in VOs There is not a single VO model: VOs may be long-term static VOs or short-term dynamic VOs. While a static VO life-time may be several months or years (corresponding to the duration of a collaboration), the life-time of a dynamic VO may be limited to the life-time of an application. A VO may be statically created by a VO administrator (static VO) or dynamically created by an application (dynamic VO). VOs also differ in their policies. A VO is created, evolves and is finally dissolved. Several kinds of modifications may happen during the VO life cycle: the addition or removal of institutions participating in the VO; the addition or removal of a site; or the addition or removal of a resource or of a member. Policies may also change over time, and the resources within a VO may experience failures. So while still belonging to the VO, these resources may be temporarily not accessible, leading to another form of dynamicity within a VO.

XtreemOS will support different VO models [26], and scalability issues can be seen w.r.t. the performance of the system, and its ability to adapt to changes. As a resource node may provide access to thousands of grid users from multiple VOs, the local operating system must still provide strong isolation properties, such as the existence or failure of an application shall not affect security and performance of applications from different VOs running on the same node).

When VOs are dynamically created or changed, e.g. when some resources fail, maintaining consistency of static local configurations becomes a complex task and heavy administrative burden, even if done automatically. Thus, in order to support large numbers of users in such a dynamic environment, implementation solutions relying on local configuration files which statically contain user/resource information should be avoided. Resources and users belonging to various administrative domains, scalability means keeping the autonomous management of user accounts and resources by the domain system administrator.

Scalability of Services XtreemOS services should be designed to scale with the number of entities (e.g. resources, users, applications, VO, sites) and their geographical distribution. On one hand, they should be fully distributed, avoid any contention points and save network bandwidth for the sake of performance. On the other hand, they should be able to securely run over multiple administrative domains. They should adapt to the evolution of the system composition coping with the dynamicity of a Grid. As a result, it will not be possible to maintain a global view of the system. XtreemOS services will also have to be self-managed

services automatically dealing with events such as a node joins, leaves, or crashes. To deal with the high churn of nodes characterizing a large scale distributed system such as a Grid, service migration should be made transparent to users. Finally, there should be no single point of failure in XtreemOS operating system. Thus, XtreemOS critical services will need to be highly available.

Scalability of application execution management To scale with the number of applications, the application execution management service will be distributed within the scope of a VO. As it is not possible to get an optimal scheduling in a large scale dynamic system where a very large number of users share resources for the execution of a potentially very large number of applications of different kinds, XtreemOS scheduling philosophy will be best effort with a job centric scheduler.

Scalability of the Grid composition service A node in XtreemOS should be able to communicate with any other node in the same VO, for example, to find resources for executing a job. As the number of nodes in a Grid may be very large, it is not conceivable that a node keeps information on all other nodes it may communicate with. For the sake of scalability, in XtreemOS, a node will only keep information on a few nodes. The intersection between information maintained by each node on other nodes belonging to the same Grid should be large enough to support simultaneous failures of multiple nodes. This means that XtreemOS will be a highly decentralized system in the same spirit as Peer-to-Peer systems that are able to cope with node dynamicity.

Scalability in data management XtreemOS data management service will be able to deal with large amounts of data stored in geographically distributed data storage units (in different administrative domains) and accessed from any Grid node. It will manage files (volumes) that are shared by Grid users from different VOs. XtreemOS will provide a Grid file system providing efficient data access and data high availability. The XtreemOS file system *XtreemFS* will perform file access control and will guarantee secure data management. To be scalable, the management of the Grid file system will be highly decentralized.

2.2 XtreemOS Overall Architecture

2.2.1 Overview

The XtreemOS project is building a operating system to support virtual organizations (VOs) in next-generation grids. Unlike the traditional, middleware-based approaches, it is a major goal to provide seamless support for VOs at all the software

layers involved, ranging from the operating system of a node, via the VO-global services, up to direct application support.

As Linux is widely available on various platforms (in particular, it is the system of choice for HPC clusters and for servers), XtreemOS implementation will be based on Linux. This should not be a limitation as Linux based OS can be executed on computers natively executing another OS by the means of virtual machines. XtreemOS integrates operating systems for the various computer architectures used in VOs, as follows.

- For stand-alone PCs (single CPU, or SMP, or multi-core), XtreemOS provides its Linux-XOS flavour with full VO support.
- For clusters of Linux machines, the LinuxSSI flavour combines VO support with a single system image (SSI) functionality.
- For mobile devices, XtreemOS provides the XtreemOS-MD flavour with VO support and specially-tailored, lightweight services for application execution, data access, and user management.

The XtreemOS project is producing various software components, ranging from Linux kernel modules to application-support libraries [40]. The overall layering of these components, grouped within *software packages*, is shown in Fig. 1. It shows all layers in the infrastructure at a very high level of abstraction. Each *layer* abstracts further from the underlying physical structure of a Grid, and consists of one or more software packages.

A software package provides one or more of the *services* of XtreemOS. Each service implements its functionality by interacting with other services in the same layer, and the layer below. Here, services can be either "classical" Grid services within the XtreemOS-G layer, or Linux extensions (kernel modules etc.) within the XtreemOS-F layer. All XtreemOS services being designed within a single project in a cooperative way, which ensures better integration and better coordination between these services than in traditional middleware stacks.

In the following section, we introduce XtreemOS key design principles to achieve both scalability and transparency.

2.2.2 Key Design Principles

VO and Security. XtreemOS supports various VO models, used in scientific as well as business scenarios. Within these models, a user can belong to different VOs, and a resource can provide computation power and storage to multiple VOs.

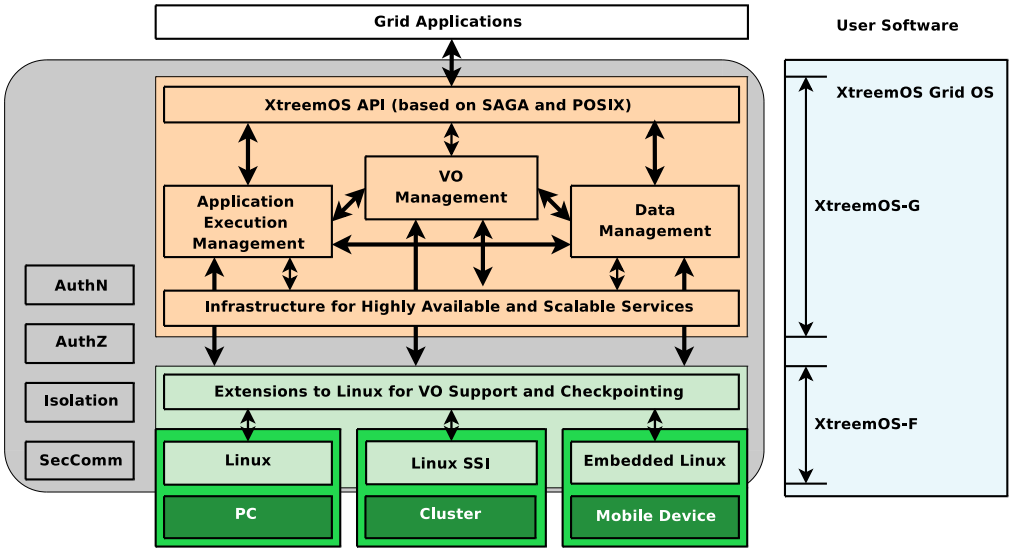

Figure 1: Layering of the XtreemOS software packages.

User management and resource management are independent in XtreemOS: there is no need to configure resources when new users are registered in VOs.

XtreemOS provides Single-Sign-On (SSO): when a user performs a "*login*" within a VO, he receives credentials recognized by all resources of the VO without any need to re-authenticate. Resource access security in XtreemOS is policy-driven: access rights to a resource are evaluated from policies provided by users, VOs and resource providers.

Detailed consideration of the key design principle for VOs and security are given in [6], with the proposed security architecture in [39, 43].

Application Execution Management. The resource discovery mechanism within XtreemOS is based on a distributed information service using P2P technology. Furthermore, services that take decisions never work with a global view of the whole system, but rather use a local viewpoint. For instance, the scheduling will not try to perform a perfect global schedule, but rather generate a job-oriented scheduling within the subset of resources obtained by the resource discovery mechanism.

To ease the use of the Grid services, it is very important to mimic the well known Linux functionality as opposed to offer different abstractions and functionality which are more oriented to the Grid. In this same spirit, reliable monitoring which can be reported to the user using familiar tools is vital to provide assurance

to users and administrators. This is a feature which is normally not found in Grid environments.

Detailed consideration of the key design principle for AEM and its architecture are given in [36, 27].

Data Management. The data management capabilities of XtreemOS are provided by the XtreemFS file system. With XtreemFS we have chosen to implement a full file system and design it for features that are expected from a grid data management system, such as federation, replication and parallel access. XtreemFS fully integrates with the VO concept and allows applications to transparently access files across the whole Grid without any further mediation by middleware layers. Being a real file system, it has full control over any access to the file data and provides real file semantics even in presence of concurrent accesses.

The Object Sharing Service (OSS) aims to ease the sharing of volatile data objects by transparently managing replicas and keeping them consistent. Grid applications can share objects through standard file system operations or by using customized functions. The latter include support for speculative transactions which alleviate network latency and avoid complicated lock management.

Infrastructure for Highly Available and Scalable Services. The infrastructure for highly available and scalable services provides generic services that can be used by XtreemOS-G services and applications running on top of XtreemOS, which underpin the resource management within XtreemOS in a scalable and transparent manner. This specifies a number of services as follows.

- **Distributed Server.** A distributed server is an abstraction that presents a collection of server processes to its clients as a single entity [38]. The address of a distributed server remains stable, even in the case of nodes joining or leaving the application. This technology is exploited in the project both as a support for highly available services (e.g., the job manager or the VO manager) and by those applications willing to make their internal distribution transparent to their clients.

- **Virtual Nodes.** A group of nodes taking part in an application can request to be organized as a virtual node. A virtual node is a fault-tolerant group where each member can take over the task of the others in case of failure [34]. Several types of virtual nodes may be provided, based on active replication, passive replication, and checkpoint/restart mechanisms provided by the XtreemOS operating system. This technology will be integrated with dis-

tributed servers to provide a single platform to support fault-tolerant, highly available services and applications.

- **Publish-Subscribe.** A common form of communication between a large number of nodes taking part in a given service or application is publish-subscribe. We will provide a fully decentralized pub/sub communication system that applications can use for their own purpose [44]. The current implementation is based on a hierarchical topic-based mode while later in the project we will evaluate if a content-based approach is also needed.

- **Resource Selection Service.** The Resource Selection Service (RSS) takes care of performing a preliminary selection of nodes to allocate to an application, according to range queries upon static attributes [33]. It exploits a fully decentralized approach, based on an overlay network which is built and maintained through epidemic protocols. This allows to scale up to hundred thousands, if not billions, of nodes and to be extremely resilient to churn and catastrophic failures. This service is invoked by the AEM service.

- **Application Directory Service.** The Application Directory Service (ADS) handles the second level of resource discovery, answering queries expressed as predicates over the dynamic attributes of the resources [25]. ADS will create an application-specific "directory service" using the NodeIDs received by the RSS, related to the resources involved in the application execution. To provide scalability and reliability, DHT techniques and their extensions to dynamic and complex queries will be used.

- **Application Bootstrapping.** Many applications need to have nodes arranged in specific overlay networks (e.g., a torus, a ring) to operate correctly. Application Bootstrapping is a set of libraries, leveraging off epidemic protocols, to make application nodes self-organize to meet the requirements [25].

XtreemOS Cluster Flavour. The XtreemOS cluster variant will be based on LinuxSSI [32, 41], which implements a full Single System Image (SSI) operating system for computing clusters. A full SSI operating system globally manages all cluster nodes resources to give the illusion that a Linux cluster is a single Linux node. The Posix interface is offered to users allowing the execution of unmodified legacy sequential or parallel applications and system administration tools. Hence, LinuxSSI makes a cluster appear as a single powerful (SMP-like) Grid node. Based on Kerrighed Single System Image (SSI) technology [17], LinuxSSI provides additional features such as a global customizable scheduler [28], the checkpoint/restart

of process trees [29], additional reconfiguration mechanisms [30], and a distributed file system [31].

XtreemOS Mobile Device Flavour. XtreemOS also provides a mobile device flavour (*XtreemOS-MD*), which fully integrates most of XtreemOS functionalities, giving users on the move full access to the XtreemOS Grid [35, 46, 45]. This kind of approach is much more *scalable* than gateway or Grid portal solutions for mobile access, as it eliminates the potential bottlenecks and single-points of failure of these gateways. This scalability factor can be specially relevant, given the enormous number of mobile devices that exist these days. Moreover, mobile Grid applications will be able to run *transparently* with little or no modifications in mobile devices, due to the inclusion in XtreemOS-MD of OGF's standard SAGA API.

Due to the current state of mobile Linux market, another key principle of the mobile flavour is *portability*. XtreemOS will provide not only a full Grid operating system for mobile devices, but also a set of open source software modules that can be easily integrated into any modern mobile Linux distribution, by avoiding excessive reliance on any specific mobile platform.

XtreemOS API In general, the XtreemOS API has to serve three classes of applications:

1. Existing Linux applications, using POSIX-standardized interfaces.
2. Existing Grid applications, using OGF-standardized interfaces.
3. New applications, using functionality uniquely provided by XtreemOS.

We have selected the emerging OGF standard *Simple API for Grid Applications* (SAGA) as the first draft API for XtreemOS. SAGA had been selected because it combines OGF-standardized API's (namely JSDL [1], BES [8], GridFTP [15], GridRPC [18], DRMAA [21]) with POSIX-like interfaces wherever possible (e.g., for files and streams). We have defined an API name space called XOSAGA (XtreemOS extensions to SAGA) that mirrors the SAGA API name space. XOSAGA contains only those packages, classes, and interfaces that require XtreemOS-specific extensions to SAGA. Together, SAGA and XOSAGA form the XtreemOS API [42, 37].

3 Comparison with Alternate Approaches

XtreemOS Grid OS can be compared to other Grid OS and to middleware approaches.

3.1 Other Grid OS

Legion [12] is an object-based wide-area operating system. Legion executes as middleware on top of individual resource operating systems but provides a uniform API and object space to users and to developers. Legion shares a large number of key requirements with XtreemOS, namely: security; a global name space; programming ease; interactivity; fault tolerance; persistence; dynamicity; scalability; and site autonomy. Being run as a middleware, Legion can manage heterogeneous resources running different operating systems. But, in return, Legion provides a specific API to users (in order to manage the global user namespace) and to developers. In contrast, XtreemOS provides the standard POSIX API to users and developers, but all computational resources must run the XtreemOS operating system. For user and resource management, Legion and XtreemOS also use different strategies. A Legion domain is autonomous and manages its local users and resources. But multiple domains can be combined in order to form larger systems: objects created in one domain can communicate with and use the services of other objects in connected domains. XtreemOS manages users and resources through virtual organizations. VOs are independent, but a user can belong to multiple VOs and a resource can provide services to multiple VOs.

Globe [23] shares many goals with Legion. Both are middleware running on top of host operating system and use class objects to abstract implementation details. But the implementation of objects is different and Globe can be presented as a distributed application environment whereas Legion is presented as an operating system for grids.

9Grid [16] is an on-going collective effort to extend the Plan9 [20] distributed operating system to Grids. The Plan9 operating system already integrates support for user authentication, resource discovery and data management in a distributed environment. Plan9 administration, limited to a single domain, is extended in 9Grid to support multi-domain namespaces and remote authentication agents. XtreemOS provides a similar functionality through the multi-VO support.

Mosix2 [2] is a management system targeted for high performance computing on x86 based Linux clusters and multi-cluster organizational Grids. Mosix incorporates dynamic resource discovery and automatic workload distribution, commonly found on single computers with multiple processors. The major limitation of Mosix2 is its weak security support: all resources must belong to the same orga-

nizational domain and must be connected through a secured network. XtreemOS also provides a single system image flavour for clusters. As it is the case for Mosix, the nodes of an XtreemOS cluster must be connected using a secured network, but the whole cluster is seen as a large SMP node in the XtreemOS Grid.

Vigne [13] provides a consistent set of integrated services (resource discovery, distributed application management, automatic application life cycle management) on top of Linux. Vigne targets scalability and transparency but does not currently provide any support for VOs, security or Grid file system as in XtreemOS.

GridOS [19] integrates basic fonctionalities common to classical grid middleware in the Linux operating system: a resource management module, a process management module, a kernel ftp server and a kernel ftp client, a high performance I/O module and a communication module. The integration of such basic functionalities in the kernel facilitates middleware support and mainly increases data management performance.

WebOS [22] provides OS services to wide-area applications, including mechanisms for resource discovery, a global namespace, remote process execution, resource management, authentication, and security. The major difference between WebOS and XtreemOS is that WebOS provides operating system services for the deployment of wide area applications whereas XtreemOS targets the exploitation of wide area resources for the execution of applications (sequential or parallel).

3.2 Grid middleware

Globus [9] has long been designed as a "sum of services" infrastructure, in which tools are developed independently in response to current needs of users. Globus provides specialised services for job submission, for file staging, for replica location and management, for publishing and querying of resource information, etc. The number of different services and the lack of consistent interfaces make a large Globus system difficult to install, manage and use. XtreemOS integrates equivalent functionalities in a more consistent way. User data are managed by XtreemFS, a shared Grid filesystem (no need to explicitly move files), automatically replicated, and accessible through the standard POSIX API. VO overlays provide dynamic resource management and selection. Globus, and grid middleware in general, rely on cluster execution services (batch systems) which do not support interactive applications. XtreemOS on its side extends the remote execution capabilities of Linux to the Grid and so preserves interactivity between the user and his jobs. Globus is the basis of a large number of production grid systems such as gLite [4] developed by the EGEE project [3] funded by the European Commission.

Some P2P based middleware have also been demonstrated as supporting Grid capabilities, but currently do not provide the whole software stack necessary to

support all needed functionalities. Zorilla [7] is a java-based middleware integrating a locality-aware P2P resource coallocation and scheduling system. Vishwa [14] exploits a two layered P2P architecture. The structured layer reconfigures the application to mask failures, while the unstructured layer reconfigures the application by adapting to the varying loads.

4 Significance of XtreemOS Approach for Users

The main goal of a new approach to Grid support such as XtreemOS is to provide real advantages to end users over conventional Grid middleware. In this section, we consider the general advantages of XtreemOS for all users and then the more specific advantages for three key classes of user: end users, systems administrators, and application programmers.

4.1 Advantages for all users

A key advantage of the system for all classes of users is XtreemOS's approach to handling heterogeneity in systems. This has three main aspects:

Heterogeneous applications. XtreemOS is designed to handle a wide range of types of application. At one end of the spectrum, large scale scientific collaborations tend to be widely geographically dispersed with a large number of institutions and last a long time, with a very general goal and relatively straightfoward security requirements. At the other end, commercial applications in business data centres typically involve a small number of partners on short time scales and tightly directed goals, often controlled by a workflow, and have key requirements for isolation and data security. By providing a sufficiently general and flexible infrastructure, XtreemOS aims to support this range of applications, which is demonstrated in the case studies within the project.

Heterogeneous platforms. XtreemOS aims to provide a single operating system which will operate on workstations, clusters and mobile devices. Thus a collaboration can work across these devices transparently, integrating a range of different platforms together within a single management system, working to common protocols.

Heterogeneous Security Systems The different security mechanisms which traditional systems use is a significant barrier to practical Grid computing. The mechanisms can be difficult or impossible to work together, and make it hard to establish

trust between entities. The XtreemOS security model uses a common global security mechanism which can be translated to work with local security infrastructures. Global security decisions are made using the common system with their local enforcement. Thus a common trust and security basis can be established while not interfering with local security policy.

We break down the XtreemOS users into a number of different groups and consider the benefits which each group would acrue from XtreemOS.

4.2 Advantages for end users

Two kinds of end users can be distinguished: users launching applications (called here application users), and service administrators (users launching applications that are in fact services such as for example a web server or a database). Some of these users will be experts, others novices or non-computing specialists. An objective of XtreemOS is to make the Grid invisible for non-expert users. For expert users, it may not be desirable to make the Grid fully invisible. The expert user may be able to provide useful information to the system for instance to optimize the execution time of his/her application. XtreemOS thus provides a number of advantages to end users.

Ease of use. A number of XtreemOS features provide a user with an easy migration from a familiar local Linux environment. The use of a Single-Sign On mechanism to the Grid within a virtual organisation allows the Grid user to access all the resources on the Grid with a single use of Grid user name and password, and then the underlying certificate mechanism will authenticate the user against resources appropriately. This provides a convenient mechanism withihout the need to manage multiple certificates explicitly. This seemless access without multiple authentication challenges also enhances the user's view of transparent access to resources, which can be called as if they appear on the local system, regardless of their actual location. Further as user commands in XtreemOS support Posix, as far as possible, one familiar interface can be used to call local and global resources.

Secure and reliable application execution XtreemOS provides a number of features which support secure and efficient execution of applications. XtreemOS provides a fine-grained control of access to resources on the available computing nodes, allowing specific data objects to be managed separately. This is supplemented by efficient application execution by identifying and utilising free compute nodes within the Virtual Organisation, so maximising the use of the available computational power, and the execution is then monitored accurately by auditing services across the Virtual Organisation.

Platform transparency XtreemOS provides a single interface which can accomodate cluster machines and mobile devices as well as conventional workstations, so it supports a ubiquitous access to services, applications and data across devices within one system. Thus end users do not need to configure their application to work on a particular platform.

4.3 Advantages for administrators

Another key set of XtreemOS users are system administrators. System administrators can include resource administrators, responsible for a particular computing resource, or Virtual Organisation administrators, responsible for supporting a collaboration within a community. Again, XtreemOS provides advantages for both groups, as follows.

Advantages for local resource administrators. XtreemOS supports the autonomous management of local resources. Security policies to access local resources, and accounts for local users, can be controlled locally by their own local administrators. Then when the resource participates in virtual organisations, these local configurations can be propagated across the virtual orgnisation and thus respected by other node. Further features which support strong isolation, including node virtualisation and virtual firewalls are being added to support high integrity resources and applications, which also satisfy the requirements of local administration on the higher security on such resources.

Advantages for Virtual Organisation administrators. XtreemOS provides a scalable virtual organisation tool, which can be controlled via Virtual Organisation roles and policies, which respect the needs of local resources. XtreemOS also considers the whole lifecycle, supporting VO establishment, change and dissolution in a single controlled environment. XtreemOS thus will give VO administrators the tools to flexibly run their VOs while having the assurance that local requirements will be respected across the Grid.

4.4 Advantages for application programmers

XtreemOS is a Grid OS which can support a wide range of applications and services. It does not assume any particular architecture beyond the basic Grid and Virtual Organisation services, many of which are transparent to the users, and is thus designed to not be limited to a particular programming paradigm. Current Linux applications can run with little or no modifications, so legacy applications

to be executed in a Grid without modification, or recompilation. It is also intended that current Grid applications will also run with little or no modifications. XtreemOS supports the emerging standard OGF standard API 'SAGA', with currently both the C and Java languages supported. As other Grid systems will also support this interface, then a Grid application defined for one system should also run any other SAGA compliant system. This would potentially include applications running across a combination of XtreemOS and other Grid middleware.

5 Conclusion

XtreemOS proposes a novel approach to the management of large and very dynamic Grid systems allowing users to transparently exploit VO-managed resources through the standard Posix interface. XtreemOS is a Grid operating system in the sense that it offers a coherent set of integrated and cooperative system services to manage jobs, data, Grid users and Virtual Organizations in a multi-domain context.

The main objective of this new approach is to provide real advantages over conventional Grid approaches. XtreemOS targets ease of use, secure and reliable application execution, and platform transparency for end users. With XtreemOS, local system administrators still manage their resources autonomously and VO administrators are provided with a scalable and flexible Virtual Organization tool dealing with the whole VO life-cycle. Transparency and scalability are the two fundamental concepts guiding the design of XtreemOS. Moreover, unlike the traditional middleware-based approaches, XtreemOS provides seamless support for VOs at all the software layers involved, ranging from the operating system of a node, via the VO-global services, up to direct application support. In contrast to many middleware approaches, XtreemOS is able to execute any kind of application from legacy Posix compliant applications to Grid-aware applications conforming to OGF standards.

XtreemOS targets both business and scientific applications. Three flavours of XtreemOS are implemented based on Linux for stand-alone PCs, PC clusters and mobile devices. The first public release of XtreemOS open source software for PCs and clusters is planned in June 2008. Further versions will then be released to refine the XtreemOS ssystem together with extensive system testing on a wide variety of case-studies.

Acknowledgement

This paper is the first white paper on XtreemOS project. The authors would like to thank all members of the XtreemOS consortium for their valuable contributions to

the ideas presented in this paper. XtreemOS is an Integrated Project supported by the European Commission's IST program #FP6-033576.

References

[1] A. Anjomshoaa, F. Brisard, M. Drescher, D. Fellows, A. Ly, S. McGough, D. Pulsipher, and A. Savva. Job Submission Description Language (JSDL) Specification v1.0. Grid Forum Document GFD.56, Open Grid Forum (OGF), 2005. `http://www.ogf.org/sf/documents/GFD.56.pdf`.

[2] A. Barak and A. Shiloh. The MOSIX2 Management System for Linux Clusters and Multi-Cluster Organizational Grids. Technical report, Hebrew University of Jerusalem, March 2007.

[3] Rüdiger Berlich, Marcus Hardt, Marcel Kunze, Malcolm Atkinson, and David Fergusson. Egee: building a pan-european grid training organisation. In *ACSW Frontiers '06: Proceedings of the 2006 Australasian workshops on Grid computing and e-research*, pages 105–111, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[4] Stephen Burke, Simone Campana, Patricia Méndez Lorenzo, Christopher Nater, Roberto Santinelli, and Andrea Sciabà. GLITE 3.1 USER GUIDE. EGEE, April 2008. Document identifier: CERN-LCG-GDEIS-722398, `https://edms.cern.ch/file/722398/1.2/gLite-3-UserGuide.pdf`.

[5] XtreemOS Consortium. Xtreemos: Building and promoting a linux-based operating ssystem to support virtual organizations for next generation grids. Technical Annex. Integrated Project (IP) in the FP6-2005-IST-5 European program, April 2006.

[6] Massimo Coppola, Yvon Jégou, Brian Matthews, Christine Morin, Luis Pablo Prieto, Óscar David Sánchez, Erica Y. Yang, and Haiyan Yu. Virtual organization support within a grid-wide operating system. *IEEE Internet Computing*, 12(2):20–28, 2008.

[7] Niels Drost, Rob V. van Nieuwpoort, and Henri Bal. Simple locality-aware co-allocation in peer-to-peer supercomputing. In *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06)*, 2006.

[8] I. Foster, A. Grimshaw, P. Lane, W. Lee, M. Morgan, S. Newhouse, S. Pickles, D. Pulsipher, C. Smith, and M. Theimer. OGSA Basic Execution Service Version 1.0. Grid Forum Document GFD.108, Open Grid Forum (OGF), 2007. `http://www.ogf.org/sf/documents/GFD.108.pdf`.

[9] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2):115–128, Summer 1997.

[10] Ian Foster and Carl Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan-Kaufmann, 1999.

[11] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *Int. J. High Perform. Comput. Appl.*, 15(3):200–222, 2001.

[12] Andrew S. Grimshaw, Wm. A. Wulf, and CORPORATE The Legion Team. The legion vision of a worldwide virtual computer. *Commun. ACM*, 40(1):39–45, 1997.

[13] Emmanuel Jeanvoine, Christine Morin, and Daniel Leprince. Vigne: Executing easily and efficiently a wide range of distributed applications in grids. In *Euro-Par*, pages 394–403, 2007.

[14] Venkateswara Reddy M, Vijay Srinivas Agneeswaran, Tarun Gopinath, and Janakiram D. Vishwa: A reconfigurable P2P middleware for Grid Computations. In *International Conference on Parallel Processing (ICPP 2006)*. IEEE Computer Society, 2006.

[15] I. Mandrichenko, W. Allcock, and T.Perelmutov. GridFTP v2 Protocol Description. Grid Forum Document GFD.47, Open Grid Forum (OGF), 2005. `http://www.ogf.org/sf/documents/GFD.47.pdf`.

[16] A. Mirtchovski, R. Simmonds, and R. Minnich. Plan 9 – an integrated approach to grid computing. In *International Parallel and Distributed Processing Symposium (IPDPS)*, April 2004.

[17] Christine Morin, Renaud Lottiaux, Geoffroy Vallée, Pascal Gallard, David Margery, Jean-Yves Berthou, and Isaac Scherson. Kerrighed and data parallelism: Cluster computing on single system image operating systems. In *Proc. of Cluster 2004*. IEEE, September 2004.

[18] H. Nakada, S. Matsuoka, K. Seymour, J. Dongarra, C. Lee, and H. Casanova. A GridRPC Model and API for End-User Applications. Grid Forum Document GFD.52, Open Grid Forum (OGF), 2005. `http://www.ogf.org/sf/documents/GFD.52.pdf`.

[19] Pradeep Padala and Joseph N. Wilson. Gridos: Operating system services for grid architectures. In Timothy Mark Pinkston and Viktor K. Prasanna, editors, *HiPC*, volume 2913 of *Lecture Notes in Computer Science*, pages 353–362. Springer, 2003.

[20] Rob Pike, Dave Presotto, Sean Dorward, Bob Flandrena, Ken Thompson, Howard Trickey, and Phil Winterbottom. Plan 9 from Bell Labs. *Computing Systems*, 8(3):221–254, Summer 1995.

[21] H. Rajic, R. Brobst, W. Chan, F. Ferstl, J. Gardner, A. Haas, B. Nitzberg, D. Templeton, J. Tollefsrud, and P. Tröger. Distributed Resource Management Application API Specification 1.0. Grid Forum Document GFD-R.022, Open Grid Forum (OGF), 2007. `http://www.ogf.org/sf/documents/GFD.22.pdf`.

[22] Amin Vahdat, Tom Anderson, Mike Dahlin, Eshwar Belani, David Culler, Paul Eastham, and Chad Yoshikawa. WebOS: Operating system services for wide area applications. In *Proceedings of the Seventh Symposium on High Performance Distributed Computing*, 1998.

[23] Marteen van Steen, Philip Homburg, and Andrew S. Tanenbaum. The architectural design of Globe: A wide-area distributed system. Technical Report IR-422, Netherlands, 1997.

[24] Wikipedia. *Cloud Computing*. `http://en.wikipedia.org/wiki/Cloud_computing`.

[25] XtreemOS Consortium. Design of an Infrastructure for Highly Available and Scalable Grid Services. Deliverable D3.2.1, December 2006.

[26] XtreemOS Consortium. Security Requirements for a Grid-based OS. Deliverable D3.5.2, December 2006.

[27] XtreemOS Consortium. Basic services for application submission, control and checkpointing. Deliverable D3.3.3, November 2007.

[28] XtreemOS Consortium. Design and implementation of a basic customizable scheduler. Deliverable D2.2.6, November 2007.

[29] XtreemOS Consortium. Design and implementation of basic checkpoint/restart mechanisms in LinuxSSI. Deliverable D2.2.3, November 2007.

[30] XtreemOS Consortium. Design and implementation of basic reconfiguration mechanisms in LinuxSSI. Deliverable D2.2.4, November 2007.

[31] XtreemOS Consortium. Design and implementation of high performance disk input-out operations in a cluster. Deliverable D2.2.5, November 2007.

[32] XtreemOS Consortium. Design and implementation of scalable SSI mechanisms in LinuxSSI. Deliverable D2.2.2, November 2007.

[33] XtreemOS Consortium. Design and Specification of a Prototype Service/Resource Discovery System. Deliverable D3.2.4, December 2007.

[34] XtreemOS Consortium. Design and Specification of a Virtual Node System. Deliverable D3.2.5, December 2007.

[35] XtreemOS Consortium. Design of a Basic Linux Version for Mobile Devices. Deliverable D2.3.3, November 2007.

[36] XtreemOS Consortium. Design of the architecture for application execution management in XtreemOS. Deliverable D3.3.2, May 2007.

[37] XtreemOS Consortium. First Prototype of XtreemOS Runtime Engine. Deliverable D3.1.3, November 2007.

[38] XtreemOS Consortium. First Prototype Version of Ad Hoc Distributed Servers. Deliverable D3.2.2, December 2007.

[39] XtreemOS Consortium. First Specification of Security Services. Deliverable D3.5.3, May 2007.

[40] XtreemOS Consortium. First version of system architecture. Deliverable D3.1.4, November 2007.

[41] XtreemOS Consortium. Prototype of the basic version of LinuxSSI. Deliverable D2.2.7, November 2007.

[42] XtreemOS Consortium. Second Draft Specification of Programming Interfaces. Deliverable D3.1.2, November 2007.

[43] XtreemOS Consortium. Second Specification of Security Services. Deliverable D3.5.4, December 2007.

[44] XtreemOS Consortium. Simulation-based evaluation of a scalable publish/subscribe system. Deliverable D3.2.3, December 2007.

[45] XtreemOS Consortium. Design of Basic Services for Mobile Devices. Deliverable D3.6.2, May 2008.

[46] XtreemOS Consortium. Linux-XOS for MDs/PDA. Deliverable D2.3.4, May 2008.